

WHAT IS CLAIMED IS:

Sub
P2

1. A method of determining whether a plurality of datapaths executing a computer program should execute conditional processing in the computer program, comprising:
determining whether PE states of all of the datapaths are disabled;
determining whether the computer program is deterministic; and
branching around the conditional processing if the PE states of all of the datapaths are disabled and the computer program is non-deterministic.

2. The method of claim 1 wherein determining whether the PE states of all of the plurality of datapaths are disabled comprises:
evaluating a processor enable bit associated with each one of the plurality of datapaths.

3. The method of claim 2 wherein the processor enable bit is enabled if it is a value of one.

4. The method of claim 2 wherein the processor enable bit is disabled if it is a value of zero.

5. The method of claim 1 wherein the determining of whether the computer program is deterministic comprises evaluating a deterministic bit.

6. The method of claim 5 wherein the deterministic bit contains a first value indicating the computer program is deterministic.

7. The method of claim 5 wherein the deterministic bit contains a second value indicating the computer program is non-deterministic.

8. A method of determining whether a plurality of datapaths executing in a program should execute a conditional processing block in the program comprising:
determining whether all PE states of the datapaths are disabled; and

Cont
A2
4 branching around the conditional processing block if the PE states of all the datapaths
5 are disabled.

1 9. The method of claim 8 wherein determining further comprises:
2 determining whether the program is non-deterministic.

1 10. The method of claim 9 wherein the branching further comprises:
2 branching if the PE states of all of the datapaths are disabled and the program is non-
3 deterministic.

1 11. The method of claim 9 further comprising not branching if the program is
2 deterministic.

1 12. An instruction set executed by datapaths during conditional processing,
2 comprising an instruction that causes the datapaths to
3 determine whether PE states of all of the datapaths are disabled, and
4 branch around the conditional processing if the PE states of all of the datapaths are
5 disabled.

1 13. The instruction set of claim 12 wherein the branching is not performed if the
2 program is deterministic.

1 14. An instruction set executed by datapaths during conditional processing,
2 comprising an instruction that causes the datapaths to
3 establish a state of PE states of the datapaths for the conditional processing,
4 determine whether the established PE states of all of the datapaths are disabled, and
5 branch around the conditional processing if the established PE states of all of the
6 datapaths are disabled.

1 15. The instruction set of claim 14 wherein the conditional processing includes an if-
2 processing block.

~~set of~~
~~state of~~

A2

- 1
- 2

Add
A2

[illegible]